

In the Claims:

Please amend the claims as follows:

1. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, said system comprising:

a markup language remote procedure call server, said server receiving said method name and associated parameters from said tokenized request markup language document;

said server invoking a method corresponding to said method name;

said server receiving return parameters from said invoked method;

said server generating a response markup language document encoded in said tokenized format containing said return parameters; and

a parser, said parser parsing said tokenized request markup language format document and presenting said document to said server such that said server receives said method name and associated parameters; and

~~wherein said request markup language document and said response markup language document are encoded in a tokenized format.~~

2. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a

request markup language format document encoded in tokenized format and transmits said document, as per claim 1, wherein said parser implements an event-based API.

3. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 2, wherein said parser translates the tokens of said tokenized request document into strings and presents said request document to said server as said strings.

4. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 3, wherein said parser translates the tokens into said strings using a code space generated offline.

5. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 2, wherein said parser presents said request document to said server as tokens.

6. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 1, wherein said parser implements a tree-based API.

7. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 6, wherein said parser translates the tokens of said tokenized request document into strings and presents said request document to said server as said strings.

8. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 7, wherein said parser translates the tokens into said strings using a code space generated offline.

9. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said

document, as per claim 6, wherein said parser presents said request document to said server as tokens.

10. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 1, said system further comprising:

a servlet, said servlet receiving a request from said client to establish a connection with said server;

said servlet invoking said server upon receiving said request, said server establishing a connection with said client;

said server registering handler objects and associated methods which were to be invoked via said tokenized request markup language documents transmitted by said client;

wherein references to said handler objects and associated methods are stored in a hash table at said server.

11. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 10, wherein, prior to invoking said method, said server determines if said method corresponding to said method name is registered with said server via said hash table.

12. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 1, wherein a transport protocol for transmitting said request markup language document is HTTP.

13. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 12, wherein said markup language is XML.

14. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 12, wherein said request markup language document is transmitted as the body of a HTTP-POST message.

15. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a

request markup language format document encoded in tokenized format and transmits said document, as per claim 14, said system further comprising:

a servlet running as an extension to a HTTP service and receiving said HTTP-POST message containing said tokenized request markup language document in the body of said HTTP-POST message;

said servlet processing said HTTP-POST message, and

wherein upon determining the body of an HTTP-POST request is said tokenized request markup language document, said servlet forwards said tokenized request markup language document to said server upon which said parser parses said document and presents said document to said server such that said server receives said method name and associated parameters.

16. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a client is passed a method name and associated parameters for a remote procedure call, marshals said method name and parameters into a request markup language format document encoded in tokenized format and transmits said document, as per claim 1, wherein said markup language is XML.

17. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, said system comprising:

a client, said client being passed said method name and associated parameters for said remote procedure call; and

said client generating said tokenized request markup language document including said method name and associated parameters and sending said document to said server; and

~~wherein said request markup language document is encoded in a tokenized format.~~

18. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 17, wherein a transport protocol for transmitting said tokenized request markup language document is HTTP.

19. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 18, wherein said tokenized request markup language document is transmitted as the body of a HTTP-POST message.

20. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 17, wherein said markup language is XML.

21. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 19, said system further comprising:

a parser, said parser receiving said tokenized response markup language document containing said return parameters;

said parser parsing said tokenized response markup language document and presenting said document to said client such that said client receives said return parameters; and

~~wherein said response markup language document is encoded in a tokenized~~
format.

22. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 21, wherein said parser implements an event-based API.

23. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 22, wherein said parser translates the tokens of said tokenized response document into strings and presents said response document to said client as said strings.

24. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 23, wherein said parser translates the tokens into said strings using a code space generated offline.

25. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 22, wherein said parser presents said response document to said client as tokens.

26. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 21, wherein said parser implements a tree-based API.

27. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 26, wherein said

parser translates the tokens of said tokenized response document into strings and presents said response document to said client as said strings.

28. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 27, wherein said parser translates the tokens into said strings using a code space generated offline.

29. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 26, wherein said parser presents said response document to said client as tokens.

30. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format

containing return parameters returned from said invoked method, as per claim 17, wherein said method name and associated parameters are passed to said client via an *invoke* method of said client.

31. (presently amended) A system for performing efficient remote procedure calls, utilizing XML as a marshalling format, where a method name and associated parameters are included in a remote procedure call request XML document encoded in tokenized format transmitted as the body of a HTTP-POST message, said system comprising:

a markup language remote procedure call server, said server receiving said method name and associated parameters from said tokenized request XML document;

a servlet, said servlet receiving a request from said client to establish a connection with said server;

said servlet invoking said server upon receiving said request, said server establishing a connection with said client;

said server registering handler objects and associated methods which are able to be invoked via said tokenized request XML documents transmitted by said client, references to said handler objects and associated methods stored in a hash table at said server;

said server determining if a method corresponding to said method name is registered utilizing said hash table and upon determining said method is registered, said server invoking said method corresponding to said method name;

said server receiving return parameters from said invoked method;

said server generating a response XML document encoded in tokenized format containing said return parameters; and

a server side parser, said server side parser parsing said tokenized request XML format document and presenting said document to said server such that said server receives said method name and associated parameters, and

~~wherein said request XML document and said response XML document are encoded in a tokenized format.~~

32. (presently amended) A system for performing efficient remote procedure calls, utilizing XML as a marshalling format, where a method name and associated parameters are included in a remote procedure call request XML document encoded in tokenized format transmitted as the body of a HTTP-POST message, as per claim 31, said system further comprising:

a client, said client being passed said method name and associated parameters for said remote procedure call;

said client generating said tokenized request XML document including said method name and associated parameters and sending said request document to said server;

a client side parser, said client side parser receiving said tokenized response XML document upon said server sending said document, and

said client side parser parsing said tokenized response XML document and presenting said document to said client such that said client receives said return parameters.

33. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format

containing return parameters returned from said invoked method, as per claim 31, wherein said server side parser implements an event-based API.

34. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 31, wherein said server side parser implements a tree-based API.

35. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format containing return parameters returned from said invoked method, as per claim 32, wherein said client side parser implements an event-based API.

36. (presently amended) A system for performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request for a remote procedure call including a method name and associated parameters in the form of a request markup language document encoded in tokenized format, said server invokes a method corresponding to said method name and transmits a response markup language document encoded in tokenized format

containing return parameters returned from said invoked method, as per claim 32, wherein said client side parser implements a tree-based API.

37. (presently amended) A method for implementing a remote procedure call in which a client generates a request markup language document encoded in tokenized format including a method name and associated parameters, said method comprising:

receiving said tokenized request markup language document, ~~said markup language document encoded in a tokenized format~~;

parsing said document to determine said method name and associated parameters;

invoking a method corresponding to said method name and passing said parameters to said method;

receiving return parameters from said method;

generating a response markup language document including said returned parameters, said response markup language document encoded in a tokenized format.

38. (presently amended) A method for implementing a remote procedure call in which a client generates a request markup language document encoded in tokenized format including a method name and associated parameters, as per claim 37, wherein said markup language is XML.

39. (presently amended) A method for implementing a remote procedure call in which a client generates a request markup language document encoded in tokenized format including a method name and associated parameters, as per claim 37, said method further comprising:

encapsulating said tokenized response markup language document as the body in a HTTP-POST message.

40. (presently amended) A method of performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request markup language document encoded in tokenized format including a method name and associated parameters, said server invokes a method corresponding to said method name and returns a tokenized response markup language document including returned parameters returned from said method, said method of performing remote procedure calls comprising:

receiving a method name and associated parameters;

generating a request markup language document including said method name and associated parameters, said request markup language document encoded in a tokenized format;

sending said tokenized request markup language document to said server;

receiving said tokenized response markup language document returned from said server; and

parsing said tokenized response markup language server to obtain said return parameter, and

~~wherein said request markup language document and said response markup language document are encoded in a tokenized format.~~

41. (presently amended) A method of performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request markup language document encoded in tokenized format including a method name and associated parameters, said server invokes a method corresponding to said method name and returns a tokenized response markup language document including returned parameters returned from said method, as per claim 40, wherein said markup language is XML.

42. (presently amended) A method of performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request markup language document encoded in tokenized format including a method name and associated parameters, said server invokes a method corresponding to said method name and returns a tokenized response markup language document including returned parameters returned from said method, as per claim 40, said method further comprising:

encapsulating said tokenized request markup language document as the body in a HTTP-POST message.

43. (presently amended) An article of manufacture comprising a computer user medium having computer readable code embodied therein implementing a remote procedure call in which a client generates a request markup language document encoded in tokenized format including a method name and associated parameters, said computer readable code comprising:

computer readable program code receiving said tokenized request markup language document, ~~said markup language document encoded in a tokenized format;~~

computer readable program code parsing said document to determine said method name and associated parameters;

computer readable program code invoking a method corresponding to said method name and passing said parameters to said method;

computer readable program code receiving return parameters from said method;

computer readable program code generating a response markup language document including said returned parameters, said response markup language document encoded in a tokenized format.

44. (presently amended) An article of manufacture comprising a computer user medium having computer readable code embodied therein performing remote procedure calls utilizing a markup language as a marshalling format in which a server receives a request markup language document encoded in tokenized format including a method name and associated parameters, said server invokes a method corresponding to said method name and returns a tokenized response markup language document including returned parameters returned from said method, said computer readable code comprising:

computer readable program code receiving a method name and associated parameters;

computer readable program code generating a request markup language document including said method name and associated parameters, said request markup language document encoded in a tokenized format;

computer readable program code sending said tokenized request markup language document to said server;

computer readable program code receiving said tokenized response markup language document returned from said server; and

computer readable program code parsing said tokenized response markup language server to obtain said return parameter, ~~and~~

~~computer readable program code wherein said request markup language document and said response markup language document are encoded in a tokenized format.~~